— course overview —

Linear Algebra and Numerical Analysis

Dr. Dipanjan Roy,

School of AIDE, Indian Institute of Technology, Jodhpur

Textbook: Gilbert Strang, *Introduction to Linear Algebra*, 5th edition, Mary Jane Sterling, Linear Algebra for Dummies, Wiley Publishing, Inc.

About Instructor for this course

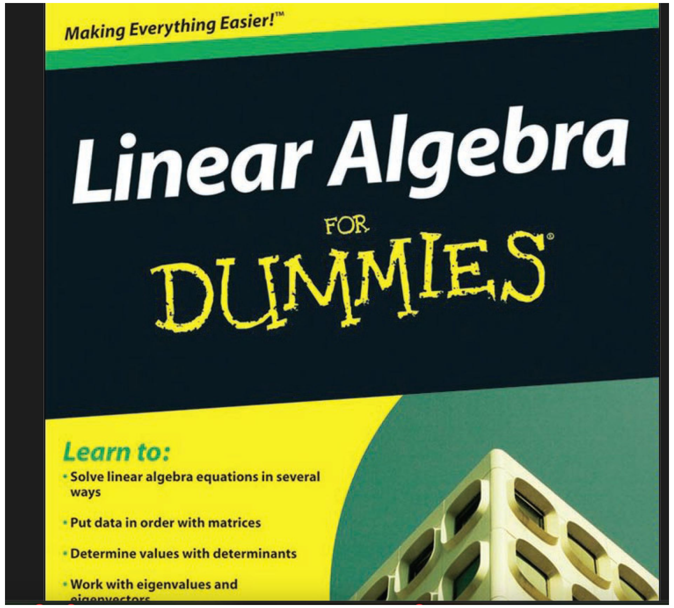
Educational Background
Ph.D. CNRS Institute of Systems Neuroscience France
Theoretical Neuroscience Group (TNG and Epilepsy Unit)
Director of Institute: Prof. Viktor Jirsa
MS in Applied Physics
Department of Physics University of Texas USA
Webpage: http://dipanjanr.com
Email: droy@iitj.ac.in

Awards:
Department of Biotechnology Ramalingaswami Re-Entry fellowship
Department of Biotechnology Innovative Young Biotechnologist Award
OHBM travel Award
CRCNS-BMBF Fellowship US-GERMANY
BCCN Fellowship Germany
Bennie Cecil Thompson Award University of Texas Best Graduate Research

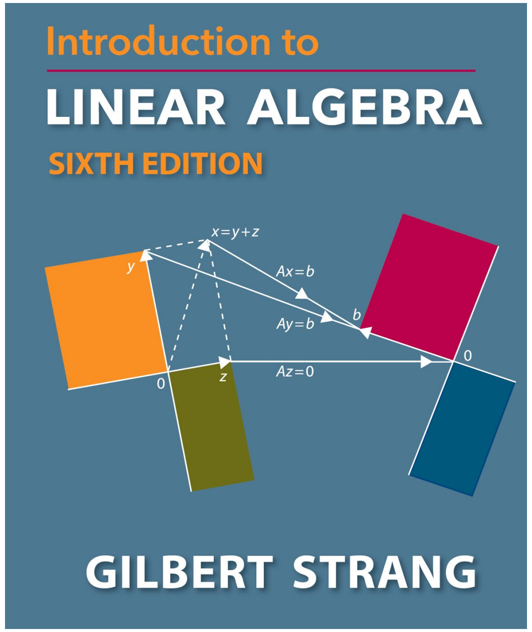
Professional experience

Associate Professor School of AI and Data Sciences Centre for Brain Science and Application Indian Institute of Technology, Jodhpur (Since October 2021)
Associate Professor, National Brain Research Centre, Manesar, India (2017 July- 2021 October)
Assistant Professor and DBT Ramalingaswami fellow Centre of Behavioural and Cognitive Sciences, University of Allahabad, India (2016 June-2017 June)
Assistant Professor IIIT Hyderabad, India (2015 January-2016 May)
Research Associate, Neurology Department, Charite Hospital Berlin, BCCN and Max Planck Institute (MPI) Leipzig, Brain Modes Group (2013 July -2014 Dec)
Postdoctoral Research Associate: Postdoctoral Research Scientist MIT Picower Center for learning and memory, McGovern Centre for Brain and Cognitive Science Mriganka Sur Lab (2011-2013), TU Berlin Neuroinformatics (2011-2013) Klaus Obermayer

The textbook



Mary Jane Sterling, Linear Algebra for Dummies, Wiley Publishing, Inc.



Gil Strang, *Introduction to Linear Algebra*, 6th edition

$$n = \mathrm{rank}(A) + \mathrm{nullity}(A) \qquad U^T U = I$$

$$A = P^{-1} D P \qquad \|v\| = \sqrt{\langle v, v \rangle} \qquad A^{-1} = \frac{1}{\det A} \mathrm{Cof}(A)^T$$

$$\det(\lambda I - A) = 0$$

$$\mathbb{R}^n = \mathrm{span}\{v_1, v_2, \ldots, v_n\} \qquad Ax = \lambda x$$

$$A^T = A$$

$$\mathrm{tr} A = \lambda_1 + \lambda_2 + \cdots + \lambda_n \qquad R = \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix}$$

Administrative Details

Lectures Recorded and live sessions (this session will have a TA).
TA for this course is Debottam Bhunia (Email: p22ai203@iitj.ac.in)
and N Akhila (Email: p23ai0001@iitj.ac.in)

TA hours during live recordings to discuss concepts of each recording and programming, Problem - solving session

Each lecture— Problem sets, Quizzes, and Questions (time-bound)
— no extensions or makeup, but the lowest problem set score will be dropped
Grading: Continuous Evaluation Classwork 15%, 3 quizzes (each 15%) (Total 60%)
& final major exam 40%

Collaboration policy: talk to anyone you want, read anything you want, but:

- Make an effort to solve a problem before collaborating.
- Write up your solutions independently.
- List your collaborators and external sources (not course materials).

Syllabus and lecture plans

- Significant overlap with Strang’s and Sterling's book: these are a useful supplement but not a replacement for attending live recordings. Likely topics:

Contents and tentative plan of topics

- Week 1-2: Monday 03/02/25. **Introduction to Matrices and Determinants:** Basic mathematical concepts related to matrices and determinants and their applications in data science. (Book: **Sterling, Chapter 1-4**)
- Week 3-5: Monday 17/02/25. **Matrix Theory I**: Matrix operations, Types of matrices, Rank of matrix, Eigenvalues, Eigenvectors, Diagonalizable matrices. (Book: **Sterling, Chapter 3-4**).
- Week 5-7: Monday 03/03/25. **Matrix Theory II:** Vector spaces R^{n}, Linear independence, Basis, Linear mappings, Affine spaces, Vector norms, Matrix norms, Lengths and distances, Angles, Orthogonality, Orthogonal basis, Orthogonal complement, Inner product, Orthogonal projections, Matrix derivatives. (Book: **Sterling**, **Chapter 4-5**.)
- Week 7-9: Wednesday 12/03/25 **Introduction to Numerical Linear Algebra:** Condition of a linear system, Eigenvalue problem, Sparse matrices, Numerical linear algebra software. (Book: **Strang, Chapter 5-6)**
- Week 10-11: 20/03/25 **Linear Solvers and Factorization Methods:** Direct and iterative methods (Gaussian elimination, LU factorization, Cholesky factorization, QR factorization, Householder’s method, Gradient descent, Conjugate gradients, Generalized minimal residual method, Preconditioning). (Book: **Strang, Chapter 1-5**)
- Week 12-13: Tuesday 01/04/25 **Computing Eigenvalues & AI Tools:** Eigenvalue computation: Direct methods, Iterative methods (Power iteration, Inverse iteration, Shifting, Deflation, Singular value decomposition). Introduction to AI tools and Python notebooks for numerical computing (**Miscellaneous sources**)

  Final exam: all of the above.

Before each exam, a definitive list of topics will be announced, along with a review session.

What is this course about?

Requires High School knowledge:

3 “linear” equations

(only ± and × constants)

in 3 unknowns

$$2x_1 + 4x_2 - 2x_3 = 2$$
$$\boxed{4x_1 + 9x_2 - 3x_3 = 8}$$
$$-2x_1 - 3x_2 + 7x_3 = 10$$

Method: eliminate unknowns one at a time.

Equivalent matrix problem

Ax = b

Ax is a “linear operation:”

A(x+y) = Ax + Ay

A(3x) = 3Ax

take “dot products” of rows × columns

$$\underbrace{\begin{pmatrix} 2 & 4 & -2 \\ 4 & 9 & -3 \\ -2 & -3 & 7 \end{pmatrix}}_{A} \underbrace{\begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix}}_{x} = \underbrace{\begin{pmatrix} 2 \\ \boxed{8} \\ 10 \end{pmatrix}}_{b}$$

A: matrix of coefficients

x: vector of unknowns

b: vector of right-hand sides

Part I: Lining Up the Basics of Linear Algebra

Linear system of equations,
in matrix form

Ax = b

$$\underbrace{\begin{pmatrix} 2 & 4 & -2 \\ 4 & 9 & -3 \\ -2 & -3 & 7 \end{pmatrix}}_{A} \underbrace{\begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix}}_{x} = \underbrace{\begin{pmatrix} 2 \\ 8 \\ 10 \end{pmatrix}}_{b}$$

Will we learn faster methods to solve this? No. (Except if A is special.) The standard "Gaussian elimination" (and "LU factorization") matrix methods are just a slightly more organized version of the high-school algebra elimination technique.

Will we get better at doing these calculations by hand? Maybe, but who cares? Nowadays, all important matrix calculations are done by computers.

Will we learn more about the computer algorithms? A little. But mostly the techniques for "serious" numerical linear algebra are topics for advanced courses. You can consider this course a prerequisite.

How do we *think* about linear systems?
(imagine someone gives you a $10^8 \times 10^8$ matrix)

- All the formulas for 2×2 and 3×3 matrices would fit on one paper. They are not why linear algebra is important (as a subfield of mathematics).
- Computers solve large problems but must be comprehensible to human beings. (And we need to give computers the right instructions and tasks to solve!)
- Understand non-square problems: #equations > #unknowns or vice versa
- Break up matrices into simpler pieces
  - Factorize matrices into products of simpler matrices: A=LU (triangular: Gauss), A=QR (orthogonal/triangular), $A=X\Lambda X^{-1}$ (diagonal: eigenvectors/values), $A=U\Sigma V^*$ (orthogonal/diagonal: SVD)
  - Submatrices (matrices of matrices).
- Break up vectors into simpler pieces: subspaces and basis choices.
- Algebraic manipulations to turn harder/unfamiliar problems (e.g. minimization or differential equations) into simpler/familiar ones: algebra on whole matrices at once

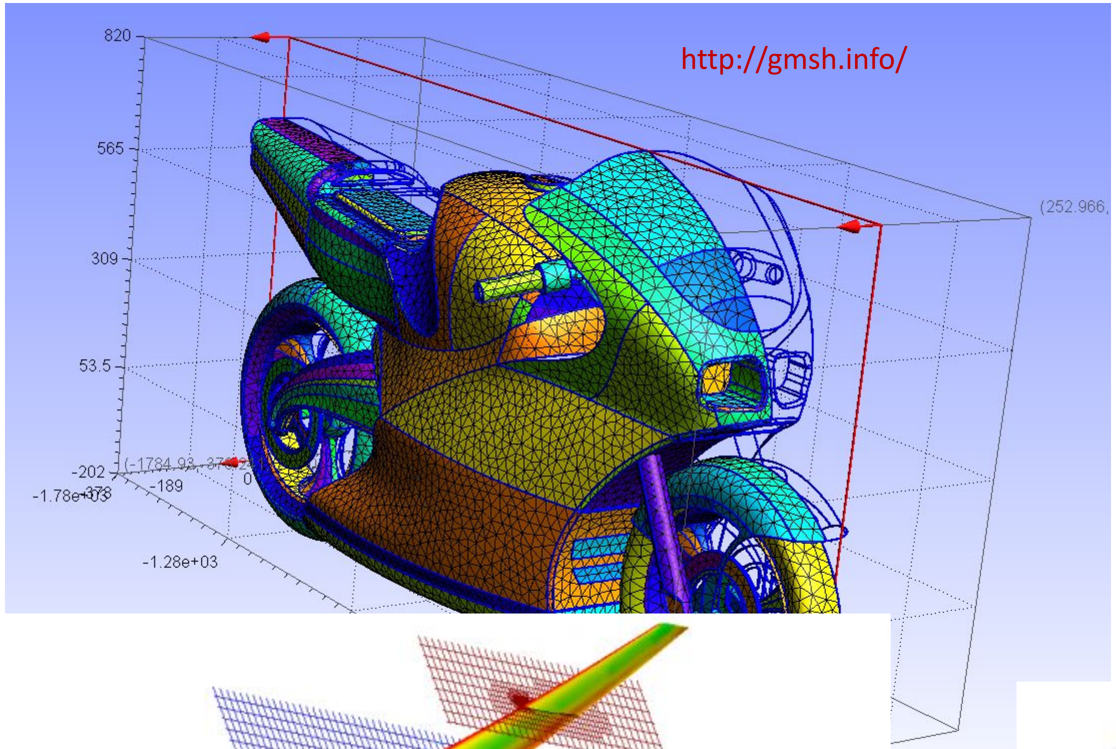
Do not expect a lot of “turn the crank” problems
on problem sets, quizzes or exams of the form
“solve this system of equations.” because that is not learning

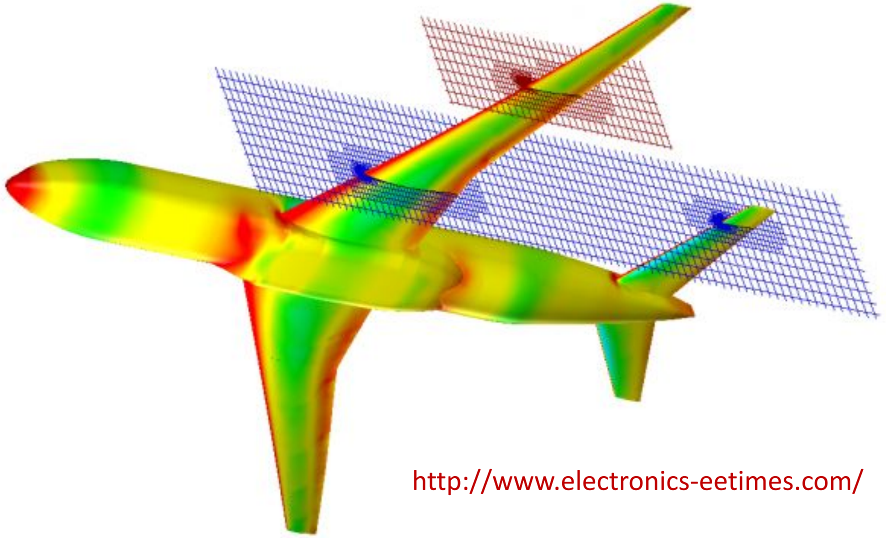
Instead, we will turn it upside-down, give you the answer to linear systems of equations, and ask about the properties of the solution from partial information, … the general goal is to require you to understand the crank rather than just turn it.

Where do big matrices come from?

Lots of examples in many fields, but here are a couple that are relatively easy to understand…

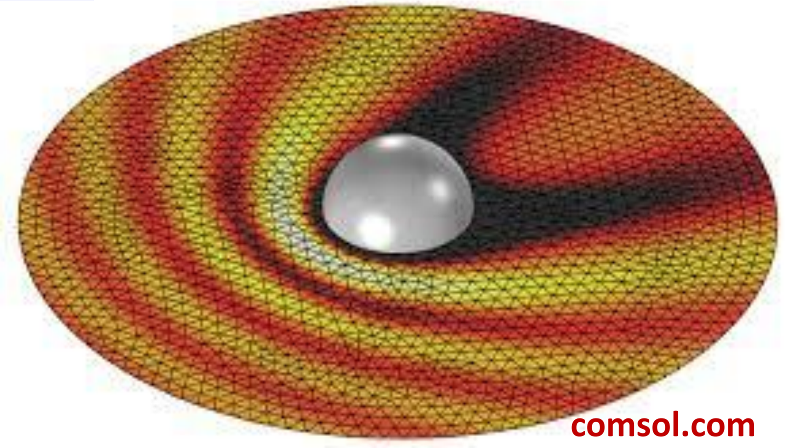
Engineering & Scientific Modeling



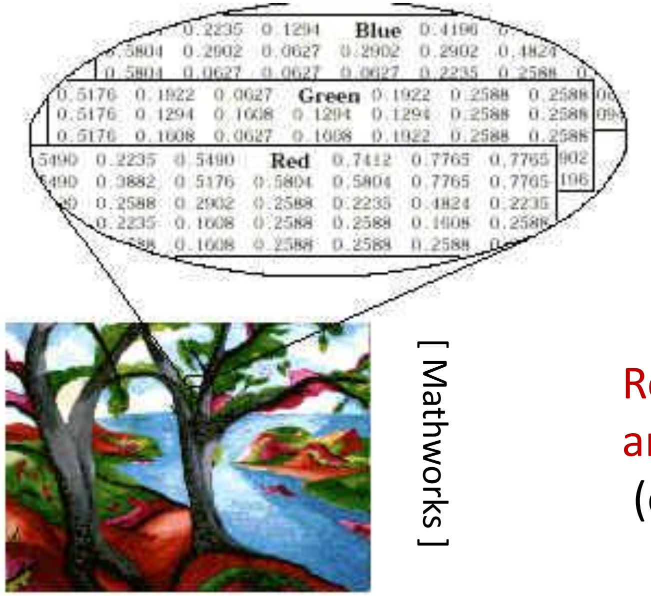


Unknown functions
(fluid flow, mechanical stress, electromagnetic fields, …)
approximated by values on a discrete mesh/grid

e.g. 100x100x100 grid
= 10^6 unknowns!



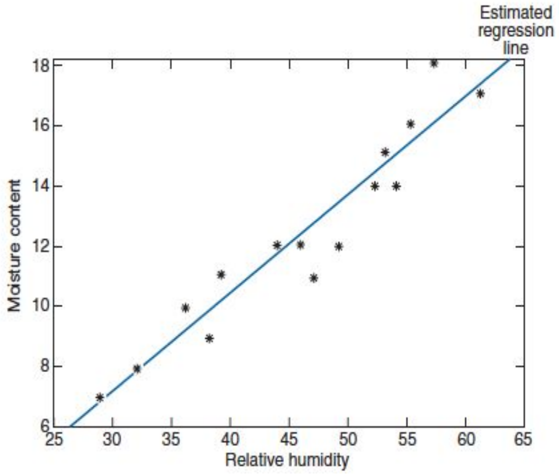
Data analysis and Machine Learning

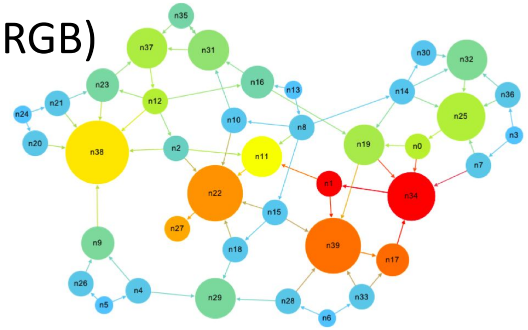


[Mathworks]

Image processing:
images are just matrices of numbers
(red/green/blue intensity known as RGB)

Regression Analysis and prediction:
(curve fitting)



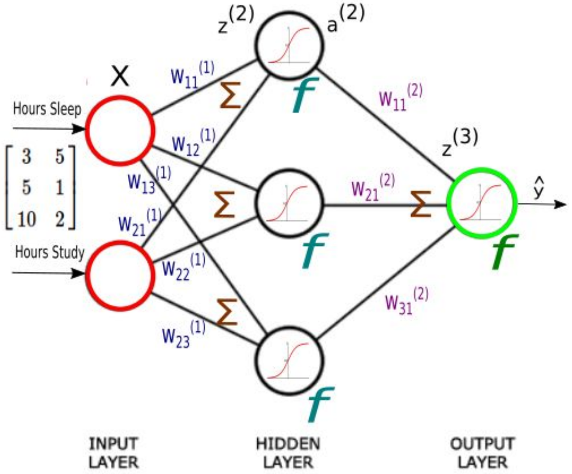


[computationalculture.net]

Google "page rank" problem
(also for genetic networks etc.)

Determine the "most important" web pages just from how they link.

matrix = (# web pages) × (# web pages)
(entry = 1 if they link, 0 otherwise)



Machine Learning

Not just matrices of numbers

- There are lots of surprising and important generalizations of the ideas in linear algebra.
- Instead of vectors with a finite number of unknowns, similar ideas apply to functions with an infinite number of unknowns.
- Instead of matrices multiplying vectors, we can think about linear operators on functions

Poisson's equation

$$\nabla^2 u = f$$

"A"
linear
operator
∇^2

"x"
unknown
function
$u(x,y,z)$

"b"
right-hand side
$f(x,y,z)$

How does this course fare against a pure math course

"applied" vs. "pure" math

few proofs vs. formal proofs expected

(deduce patterns from examples, informal arguments)

(definitions to lemmas to theorems … training in proof writing)

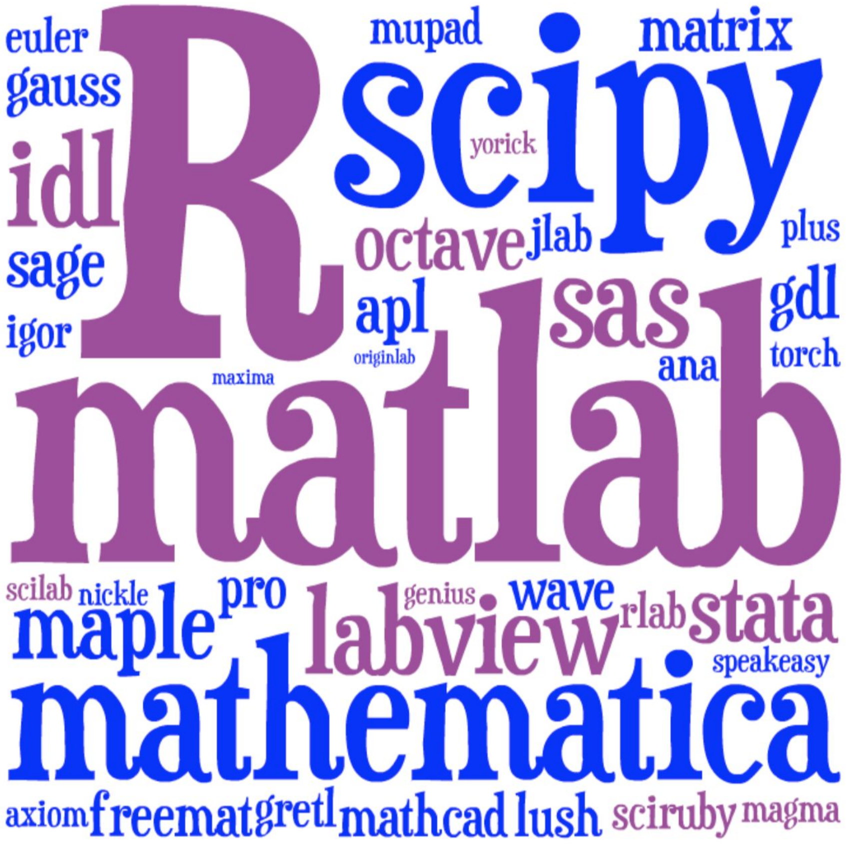
more applications vs. more theorems

more concrete vs. more abstract

some computers vs. only pencil-and-paper

Computer software

Lots of choices:



[image Credit: Viral Shah]

One can also consider using
a relatively new language
that scales better to real problems.

julialang.org

No programming is required for this course,
just a "glorified calculator" to turn the crank.

Most of the problem sets will be using ipython
Notebook and Jupyter to turn the crank
one can Use julia online: log in at juliabox.com
see "Julia" link on Stellar